Single Sign-On (SSO) - OIDC

Introduction:

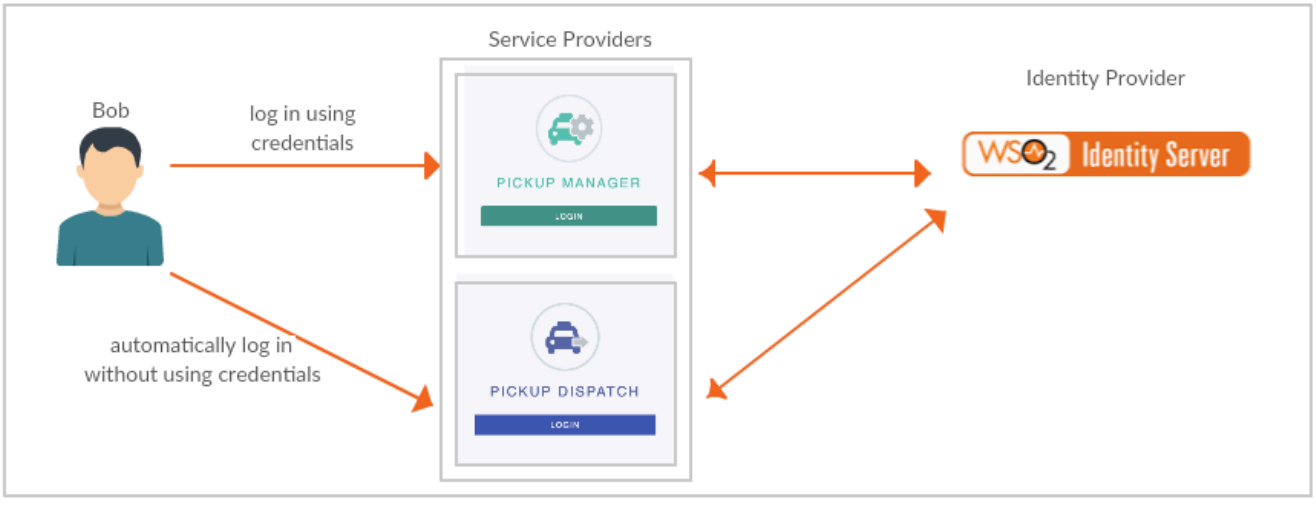
Pickup is a cab company that has many employees and different internal enterprise applications. Following are two such applications:

- **Pickup Dispatch**: This application helps manage the overall operations at Pickup.
- **Pickup Manager**: This application helps allocate vehicles to drivers.

Pickup is using **WSO2 Identity Server** as the identity provider for their applications.

Bob is a Pickup employee who usually forgets application passwords. As a result, Bob uses the same credentials for all the Pickup applications. However, due to busy schedules, Bob does not like entering the credentials at every login. So, the WSO2 Identity Server team suggested Bob to use Single Sign-On (SSO).

With SSO, Bob only needs to provide the login credentials to one Pickup application and automatically be logged in to other Pickup applications.



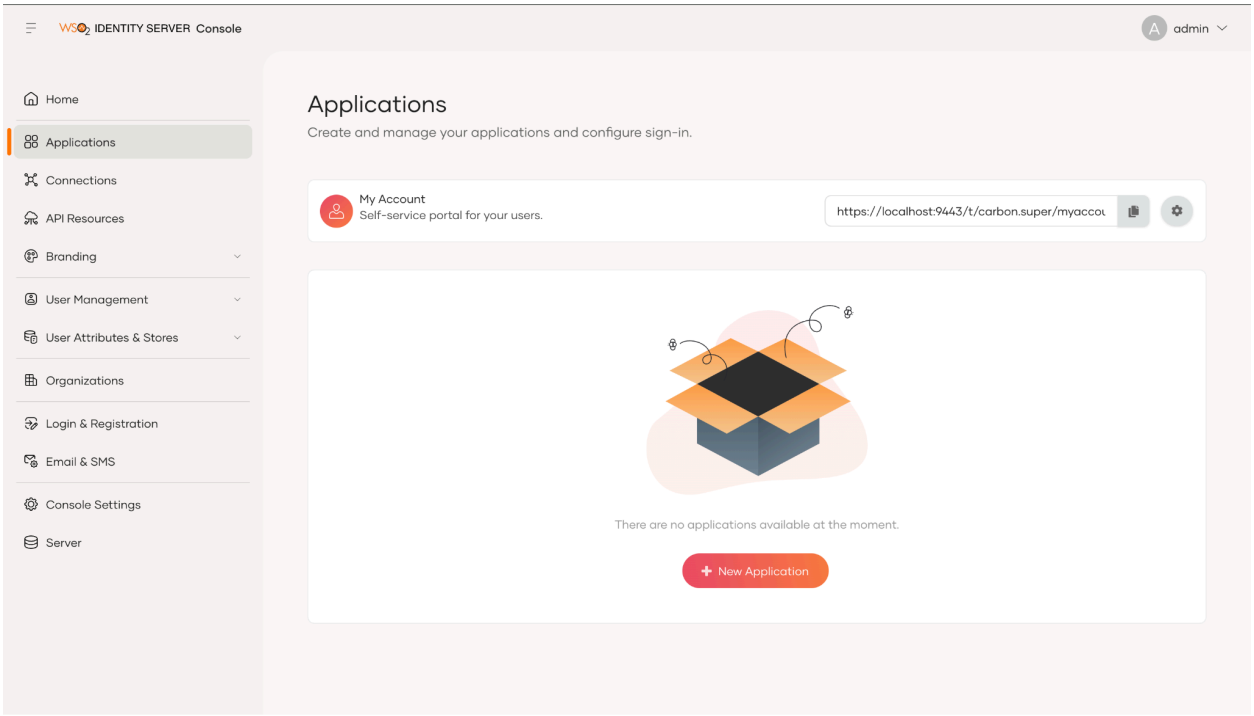
This tutorial will allow you to have hands-on experience on how to configure **SSO** with WSO2 Identity Server using **OIDC protocol**.

Setting up:

1. Download a tomcat server 8 or 9, run the server on port 8080.
2. Download the following two war files from the repository by clicking on them:
    - Pickup-dispatch.war
    - Pickup-manager.war
3. Deploy them in the tomcat.

Configure Service Providers

1. Login to the WSO2 Identity Server **Console,** using your admin credentials (e.g. admin:admin).
2. In the WSO2 Identity Server **Console,** from the menu click **Applications.**



4. Click **New Application.**
5. From the given set of templates select **Traditional Web Application** template.
6. Fill the fields in the create application wizard.

    Name: *Dispatch*

    Protocol: *OpenID Connect*

    Authorized redirect URLs:

    http://localhost.com:8080/pickup-dispatch/oauth2client

    ***Note**:*

    The callback URL is the Application's URL to which the authorization codes are sent. Upon successful authentication, the browser should be redirected to this URL.

7. Click **Create** and note the **Client ID** and **Client Secret** created.

8. Similarly, create a service provider for the Pickup Manager application with the following:

   Name: *Manager*

   Protocol: *OpenID Connect*

   Authorized redirect URLs:

   http://localhost.com:8080/pickup-manager/oauth2client

9. Edit the consumerKey and consumerSecret fields in `dispatch.properties` file in `<TOMCAT_PATH>/webapps/pickup-dispatch/WEB-INF/classes` with the copied **Client ID** and **Client Secret** in step 7 above.

   Verify the below properties are in order.

```
consumerKey=<ClientID>
consumerSecret=<ClientSecret>

callBackUrl=<Assertion_consumer_URL>
authzGrantType=code
scope=openid internal_application_mgt_view

enableOIDCSessionManagement=false
enableOIDCBackchannelLogout=true
authzEndpoint=https://localhost:9443/oauth2/authorize
OIDC_LOGOUT_ENDPOINT=https://localhost:9443/oidc/logout
sessionIFrameEndpoint=https://localhost:9443/oidc/checksession
tokenEndpoint=https://localhost:9443/oauth2/token
claimManagementEndpoint=https://localhost:9443/services/ClaimMetadataManagement
Service
post_logout_redirect_uri=<Assertion consumer URL>
api_endpoint=http://localhost:39090/bookings
adminUsername=admin
adminPassword=admin
```

10. Similarly edit the consumerKey and consumerSecret fields in `manager.properties` file in `<TOMCAT_PATH>/webapps/pickup-manager/WEB-INF/classes` with the copied **Client ID** and **Client Secret** in step 8 above.

11. Restart the tomcat server.

Try It:

Follow the steps below to try out the sample applications:

1. To access the Pickup Dispatch application, go to the following URL.

   `http://<TOMCAT_HOST>:<TOMCAT:PORT>/pickup-dispatch`

   ex: http://localhost.com:8080/pickup-dispatch
2. Sign in using Tommy's credentials.
3. To access the Pickup Manager application, go to the following URL.

   `http://<TOMCAT_HOST>:<TOMCAT_PORT>/pickup-manager`

   ex: http://localhost.com:8080/pickup-manager
4. Note that Tommy will be automatically logged in to the Pickup Manager application.

You have successfully configured OIDC-based SSO using WSO2 Identity Server as the identity provider.